>>>> TRAKCESS <<<<

Brief documentation for the Preliminary Version 0.0

ROXTON BAKER
56 South Rd., Ellington, Ct. 06029
(203) 875-2483

Congratulations! You have just swapped for, been given, or found under a rock a prelim copy of THE MOST POWERFUL TRS-80 disk access utility yet written. I hope that you will have a chance to try it out and test its features and will let me know of any suggestions you have or problems you find. Your input will be appreciated.

INTRODUCTION

Several fine disk access programs are currently available, such as Superzap, Z80ZAP, and some newer ones. For zapping standard TRSDOS disks they are excellent, featuring quick access to a file's sectors and convenient "paging" through them. Trakcess is not intended to replace these utilities. It is unable to interpret a disk's directory, and since it makes no distinction between "standard" sectors and any other, it cannot offer the pre-formatted display and easy file I/O of, say, Superzap. Trakcess was written with different goals:

- * To bring under direct user control, with as much software assistance as is practical, every capability of the TRS-80's 1771 Floppy Disk Controller (FDC).

- * To combine these capabilities into powerful "intelligent" functions whose only limitations are those of the machine in general, or of the FDC in particular.

The inspiration for this work was Bill Barden's excellent and unique "Disk Interfacing Guide for the TRS-80" which will hopefully, upon the advent of Trakcess, find renewed interest among serious disk users. If you don't have it, contact The Alternate Source (see below). They will be handling Trakcess and may arrange with Bill to sell his book as well. Certainly you will need it if you are not already an expert on the FDC. This preliminary documentation will assume in many places that you understand the elements of track formatting and sector identification. AT A MINIMUM the non-expert will require a copy of the Western Digital data sheet on the 1771 FDC. This can be found in the Expansion Interface Tech Manual. Or you may persuade WD to send you one. National Semiconductor, which is a second source for the 1771, also has a useful data sheet on it and a good App. Note (#220).

Trakcess is powerful precisely because it works on the most elementary levels. It has been padded around with lots of interactive software and utility functions so that it remains fairly efficient to use, but it is not the tool of choice for simple zapping. Trakcess assumes very little about the disks it is used on. As a result there are often many questions for the operator to answer. To make full use of Trakcess you must be prepared to study the literature and experiment with the possibilities. I recommend taking a few track dumps of disks you have on hand and comparing what you find there to what the books say. Then try making up your own tracks and formatting a blank disk with them. Trakcess puts all of this at your fingertips. The real fun, I think, is when with a little practice you tackle a "protected" disk and crack it. Some are really tough, while others have only a single sector's ID changed. Trakcess has a number of features that will assist you in this, such as (S)can Track Sectors and (L)ocate Disk Sectors. A printer, which these commands support, will be a big help. These commands embody some routines that are simple conceptually but that were quite difficult to write. They are far more potent than they seem and form the heart of the highest level commands in Trakcess, such as (C)opy Track and (D)uplicate Disk. These latter are capable of analyzing many widely different disk formats and duplicating them, and are far removed from the primitive FDC functions. However, in the event that they require your assistance, you will still have to know what you're doing.

USING TRAKCESS

This is a preliminary copy and these are preliminary instructions. I will briefly discuss the menu commands as they exist today (Sunday), and mention a few fine points. I assume the reader is familiar with disk structure and formatting, although not all the commands require such knowledge. I HIGHLY recommend that you obtain the references mentioned above to consult when necessary. Trakcess requires a 48K machine. The (C) and (D) commands require two drives. Trakcess consists of /CMD file and a Basic file. To get started from DOS, just type 'TRAKCESS'. Then follow the prompts.

1) Select Drive

You must tell Trakcess which of the four drives is to be the currently active one. You'll be asked its head position. Normally, you won't know, so just press <Enter>. Or enter "0", which will restore the head to track 0 (the outermost one). If the drive has been previously selected, Trakcess will remember its position when you press <Enter>. Note that no other disk-related commands will work until a drive has been selected!

Trakcess contains all of the disk I/O routines that it requires, so once it is running you do not need a DOS disk in drive 0. However, if Trakcess stops on a program error, put a DOS disk in drive 0 before continuing in case BASIC looks for one. Otherwise you may hang the system.

2) I and O

These step the head of the selected drive In and Out one track. "In" means closer to the center of the disk. These two keys repeat. Trakcess assumes thirty-five tracks. This is defined by DM (presently set at 34) near the beginning of the program. If you wish to work with a different number of tracks, change DM to that number minus 1.

3) (G)o to Head Position

Allows you to quickly position the head of the active drive at any track. Enter the desired track location in decimal.

4) (R)ead and (W)rite Sectors

These bring up an important point in the disk formatting: REGARDLESS of what track a sector is actually, physically on, it must be identified by two, two-digit hex numbers and a type flag. The first number is the TN. The second number is the SN. These DO NOT necessarily have any relation to the actual location of the sector on the disk! If you never think of them as anything other than arbitrary hex numbers with values 00 to FF, you won't get confused. These are the first things that people mess around with when "protecting" their software. Unlike most other disk utilities, Trakcess doesn't care whether these numbers are normal or not - and neither should you. The only restriction is that on a given track all sectors must have a different TN,SN pair. Otherwise sector reads will only pick up the first one.

The "type" flag is also important. It denotes whether the sector is an IBM-standard length or not. The allowable IBM lengths are 128, 256, 512, and 1024 bytes. A "non-IBM" (I call them NBM) sector may be any multiple of 16 bytes. A mini-disk track can only hold about 3120 bytes, so you probably won't see sector lengths beyond this. Although it is not required when reading or writing sectors, you should be aware that each sector has a third number called SL which defines its length. SL is 00-03 for IBM sectors, choosing one of the four lengths above. SL is 00-FF for NBM sectors, in which case it denotes the length in multiples of 16 bytes. The SL is mentioned again later, under (B)uilding format tracks. Some software virtuosos are now putting NBM sectors on their disks to keep you from making copies for your friends, who after all DIDN'T PAY FOR IT, and thus in the eyes of God, man, and all other disk utilities, should do without. Trakcess, I am mortified to reveal, eats up NBM sectors.

A similar story involves the Data Address Mark, which must be specified when a sector is written. The standard DAM is FB, except for TRSDOS directory track sectors which get an FA. At least one vendor, whom I shall not embarrass by name, has started putting F8 DAM's on his Adventure #9 disks. The DAM's are not important as far as reading or moving sector data is concerned; Trakcess will tell you what they are and duplicate them if you wish. For more information on them, consult the 1771 data sheet.

One last aspect of sector (R)eading is in the recovery of lost data. Trakcess will report an unsuccessful read and will give you the option of infinite retry. Unlike Superzap 3.0, Trakcess does not step the head all over the disk between attempts. That seems so hard on the drive. All sector (W)rites are verified by rereading, so data transfers under Trakcess are reliable.

5) (T)ake and (P)ut Tracks

These two commands involve the transfer of a whole track (about 3120 bytes) of information. A track read followed by a scan with Trakcess' editor will show you literally everything that is on the track. The track write is normally used only for formatting a blank disk with empty sectors. You can create any format track you want with the (B)uild command, then write it to disk with (P)ut. If none of the sectors on a track contain F7 bytes, it is feasible (though not always successful) to (T)ake a track from one disk and immediately (P)ut it onto another. Synchronization is a problem here, but it can work. In fact, sectors with F7 bytes can then be transferred separately using (R) and (W). Trakcess will do these things, but considers them poor form.

6) (S)can Track Sectors or (L)ocate Disk Sectors

Trakcess will search the current track (or the whole disk) for all useable sectors. It will determine all the important information for each (TN,SN,SL,IBM/NBM,DAM) and will present it to the screen or printer. Try this command on a standard TRSDOS disk, a CP/M disk, a TBS Security disk, the Microsoft Adventure disk, or even an Intecolor disk if you have one, and compare the results. If there are any "false" sectors (without a data record following, or with a bad ID pack) Trakcess will note the fact. If you are attempting to recover a damaged disk, you may be able to find some sector data using a track read, and transfer it to a track you have built using the (B)uild format track command discussed below. As yet software vendors have not tried using such sectors for "protection", but eventually someone will do so. Should you ever need to duplicate such a disk, use the (B) function and the editing technique outlined there.

Note that the (L)ocate command first tells you which tracks even have sectors, and then gives you the option of a full report.

7) (C)opy Track

You are prompted for a drive and track to copy to. You cannot copy to the active drive; another must be specified. Trakcess will search the current track and will identify all the sectors on it. Then it will build a matching format track in memory and write it to the target disk. Finally, it transfers (with verify) all of the sectors. You are given the option of using the existing DAM's or specifying a new one. If any sectors on the source track are damaged or of indeterminate length, you will be prompted for the length you wish to make them. To decide on this, do a track read and look at that sector in memory. Try choosing the next larger multiple of 16 bytes than the sector's apparent length, unless you know what it should be.

8) (D)uplicate Disk

This command does nothing more than set up both drives and repeat the (C) command for each track. Any disk that is not REALLY tricky can be duplicated automatically in this fashion, but it takes about 13 minutes. It's a good idea to check out the target disk first by formatting it using TRSDOS, and then bulk-erasing it. Note that unlike Superzap et. al., Trakcess copies to a blank disk rather than to a formatted one. If any sectors won't copy, note them and proceed. You may later be able to use (R) and (W) to transfer them, or that entire track may be retried with (C).

The (D) command does require you to swear that the copy you're making is for your personal use only, and that it will be used only on the same machine as the master copy was purchased for (excepting parts changes for certifiably defective components), and that the master will not simultaneously be used on another machine, or subsequently be sold to another individual without this copy being destroyed. Simply hold a notarized affidavit to this effect in front of the screen, while pressing (D).

9) (E)dit or Fill Memory

This command allows you to edit memory with the scrolling, Electric Pencil-like editor utility that is built into (and was written especially for) Trakcess. Or, you can fill memory between specified addresses with a specified byte. The editor requires a four-digit hex starting address. At the address prompt you may just press <Enter> to leave the editor. To return to the address prompt at any time, use shift/left-arrow. On the right is continuously displayed the current address of the blinking cursor, and below that a "reference address" and the number of bytes (dec.) that the cursor is offset from it. Pressing shift/break will reset the reference address to the current cursor address. The arrow keys move the cursor; shift/up-arrow and shift/down-arrow scroll rapidly. Pressing shift/clear will change the type of dump from hex to ASCII and back again. Under either mode, whatever you type in will be put into memory at the cursor location. A little practice will clarify these instructions. The editor is much easier to use than to explain.

10) (B)uild Format Track

This powerful command allows you to specify, interactively, a track to be created. There are no restrictions other than overall length on the track contents. You may specify up to 80 different sectors, of differing lengths and types, with any names. You are allowed to change the specifications on any sector; when finished, the format track may be created (at DC00H+) and (using P) written to any track of the disk. This command is very useful in creating disks that can more efficiently store and retrieve the types of data you wish to work with. For example, you might create a track with only one sector of 3024 bytes. This entire block could be read into RAM quickly using a short sector read routine as explained in Barden's book. You might also use (B) to hand-tailor a track to match one on a "protected" disk. Like many other features of Trakcess, this one can also be used to create tricky disks. I cannot, of course, be responsible for such abuses.

To specify a sector you must supply a TN, SN, SL, and must indicate (I)BM or (N)BM type. A typical entry would be:

01,2D,03,I

This specifies a sector with TN=01, SN=2D, SL=03, and of IBM type. Correlating the IBM type and the SL value indicates that the sector will be 1024 bytes long. The next sector might be specified as:

C1,77,42,N

resulting in a sector with TN=C1, SN=77, SL=42, and not of IBM type. Since it is an NBM sector, its length is calculated by multiplying the SL value (which is in hex) by 16 bytes. This sector would be 1056 bytes long. These two sectors may or may not be the only ones on this track.

Once you have created the format track at DC00H+, you may wish to edit it directly in memory before writing it to disk with the (P) command. Do this by returning to the menu and using the (E)dit function to work at DC00H. One reason you might want to do this is to zero out the DAM for a short (say 16 byte) sector that you have created, thus generating a "false sector ID"; that is, one with no data after it. Or you might wish to take out the F7 bytes that are put into the format track after every ID pack and sector record. This F7 is what causes the FDC to write two CRC bytes to the disk when it is encountered. Taking it out will result in a false ID pack (if you take out the ID CRC) or an always-bad sector (if you take out the sector CRC). Such unproductive shenanigans will only be necessary if you are breaking a "protected" disk. Hopefully you do not patronize vendors of such software and will never need these tricks. I mention them just in case you get stuck. People selling "protected" software NEVER advertise the fact, because they know very well it would decrease their sales. They prefer to deceive you until they have your money. Then they'll argue that "protection" is proper, and that the customers don't mind - but do they say it where it counts, in their ads? Not one of them! Rank hypocrisy is what it is, and they do not deserve your business.

The (B) command is probably the most fun thing Trakcess offers, next to stepping the head back and forth.

11) (F)igure CRC's

This command will allow you to calculate the two CRC bytes for any block of code in memory, or for any bytes you type in. This is useful when examining damaged or tricky sectors or sector ID's. The CRC algorithm used by the FDC is the IBM SDLC standard. It it sufficient to consider the CRC as a two-byte checksum of the data it's applied to - a checksum that is always initialized to FFFF. The FDC starts calculating a CRC whenever an Address Mark of any kind (F8-FE) is encountered during a track write, and whenever a sector is being written or read. A subsequent F7 (on a track write), or the end of the sector, cause the FDC to spit out the two current CRC bytes. On any write, these go to the disk. On a sector read, they are reported back to the CPU, along with an indication of their correctness (as compared to the CRC bytes on the disk). Consult the references for more details on these sequences.

12) (H)ex dump to the printer

This actually gives you a choice of hex or ASCII dump, starting at any address. Like all the other printer options in Trakcess it will only be offered if you have said that a printer is ready. It is supposed to pause when the space-bar is pressed, and break to the menu if 'M' is pressed. My printer requires so much software to drive it that I can't be sure if this feature is working correctly or not. Please let me know if you have trouble with it.

There is more that can be said about Trakcess. It does have a lot of error-checking built into it on operator inputs, so you should never accidentally crash it. There must still be many bugs in it and I'm sure you'll discover a few. Drop me a postcard if you do.

At almost every request for input, pressing or entering 'M' will abort the current command and take you back to the menu. Also, lengthy operations can usually be terminated by holding down 'M'. More than two-thirds of Trakcess' code is devoted to checking for the 'M' key. Most of the remainder goes toward supplying, at <Enter>, the default value I judge most desirable for each operator input.

Trakcess should hit the market in September. I hope to have added a few more features to it by then, and to have greatly strengthened the (S)can routine against tricky sectors. There is still plenty of room for improvement. If you have any suggestions, call or write. As mentioned earlier, the final (!) version of Trakcess will be available from:

The Alternate Source
1806 Ada St.
Lansing, Mich. 48910
(517) 487-3358.

TAS tries their best to offer good software at reasonable prices, and they work hard for their customers. Check them out first when you are looking for a program. They may well have a lower price, a package deal, or some free programs to throw in. You should also consider a subscription to their magazine, which often rivals 80-US in useful content, and costs much less ($9 for 6 issues). I hope to have occasional pieces on the uses and fine points of Trakcess in future issues. This depends in part on how popular the program is. TAS will be going out on a limb to offer Trakcess under a unique arrangement that will let people share the cost (about $25) with a few friends. If you appreciate such an effort, support it by getting together and ordering at the time. It is not impossible to start a trend. And tell other people about TAS and Trakcess. A tiny ad can't say much. Big ads can sell anything, but at over $1000 apiece they drive prices way up. I should also mention that the many personal opinions I express in this documentation DO NOT IN ANY WAY represent the position of TAS. They also sell software written by authors with totally different views, and they don't take sides.

With that in mind, I have a few words about "protected" software. I don't like it, because it always makes modifying or moving the code difficult, and it is usually a pain to backup and use. People do it because they think that their sales will be hurt by copies being passed around. ,Some software vendors (who will not hesitate to deceive you, see above) get extremely righteous on this subject. Rabid, in fact. I say that with 300,000 (or more) TRS-80's out there, and the fact that relatively few owners are in close contact with large numbers of other users, and the fact that most owners are quite willing to spend SOME money on software, there will be plenty of sales potential even after all the swapping. And I challenge those same vendors to say right in their ads that "This software CAN'T be copied - so don't expect to!". Let them forego all sales to groups of people who try to share the cost of software. Let them take the bad with the good. A vendor has every right to "protect" his software - as long as he's honest about it.

I applaud Randy Cook's decision not to "protect" his new VTOS 4.0. He has lost a year to Apparat in the DOS aftermarket largely because his VTOS 3.0 was viewed, not unjustly, as a very strange beast. This impression was often gained by watching 3.0's gyrations on boot-up - checking for passwords and all. If that didn't do it, making a "system disk" from the "master disk" would. Mr. Cook is certainly an expert with the DOS and his "protection", after a year, remains the best effort yet. In spite of that it was stripped off in an evening, by amateurs, long before Trakcess was available to help. So what did it get him? A bad name, that's what - and if ten people in the nation bought his DOS because they couldn't copy their pal's, I'd be surprised.

Contrast this with Apparat's apparently successful marketing of a non-protected DOS. Widely swapped, it is also widely purchased. People are banging down Apparat's doors for Newdos/80. Everyone knows how good Newdos 2.1 was, and it's NOT because everyone bought it.

There is no conclusion; each case is different. I am proud to say that of the hundreds of hours I've spent on Trakcess so far, not one second has been wasted on making it hard to copy. I hope that Trakcess will eventually reach everyone who wants it. This will ONLY happen if TAS is able to continue marketing it. The best course of action is: when you find software that you genuinely think is good and a good value, TALK IT UP. Tell your friends and contacts. Even more importantly, write in to magazines. The immense size of the market out there will take care of the rest, and everyone will be better off for it.

Although I have not yet gotten around to Trademarking (TM) every little program phrase and feature, you will notice the copyright statement on Trakcess and this documentation. It is included solely to protect this work from being stolen and sold for money by anyone other than myself or my agents. That to me is a true crime. If we didn't have to worry about such things more software could be sold under lenient "copying" terms. Many authors are unable to distinguish between software swappers and real program pirates - so they lump them together. That's easy and requires no thought, and thus is consistent with the ways our laws are often generated and interpreted. Why, in fact, should an author care if someone uses his software without having individually bought a copy? He should really be concerned with staying in business and being compensated for his efforts. If he's achieving that, then what the hell? It is merely convenient for him to point to the non-paying user as a threat to his sales, and the copyright law is equally convenient to brandish about and wax moral over. His fellow authors encourage such an attitude and make the same noises themselves. This costs them nothing and distracts attention from questions of software quality and value for money. It also leads to absurd statements like: "My $15 software has been copied 5,000 times so I've lost $75,000". A friend ox

STOCKTON TRS-80 ABUSERS GROUP
6362 Savannah Place
Stockton, CA
95205